

IN THE CLAIMS:

The following is a complete listing of the claims, and replaces all earlier version and listings.

1. (currently amended) A method of generating an encoded representation of ~~parsing~~ a markup language document comprising syntactic elements, said method comprising, ~~for one of said syntactic elements~~, the steps of:

parsing the markup language document to identify at least one syntactic element of that document;

identifying a type of the element;

processing the element by applying ~~determining~~ a hash function thereto ~~representation thereof~~ if said the type is a first type, the hash function generating a numeric code from the element; and

generating ~~augmenting an at least partial structural representation of the document using the encoded hash representation including the numeric code~~ if said the type is said the first type.

2. A method according to claim 1, wherein said the parsing is event-based parsing.

3. (currently amended) A method according to claim 1, wherein said ~~hash representation~~ the numeric code is determined using one of:

a hash algorithm;

a first reference to ~~said~~ the hash algorithm dependent upon an associated Universal Reference Indicator;

a second reference to ~~said~~ the hash algorithm dependent upon an associated namespace; and

a third reference to ~~said~~ the hash algorithm dependent upon an associated Extended Markup Language declaration[[:]].

4. (currently amended) A method according to claim [[2]] 1, wherein ~~said~~ the first type is one of:

one of a structural element and a part thereof;

a definition of ~~said~~ the structural element;

a declaration of ~~said~~ the structural element; and

a match for ~~said~~ the structural element.

5. A method according to claim 4, wherein ~~said~~ the structural element is a tag.

6. (currently amended) A method according to claim [[2]] 1, wherein the ~~hash representation~~ numeric code is a unique code for ~~said~~ the one syntactic element, ~~said~~ the element having less than a first number of characters.

7. (currently amended) A method according to claim [[2]] 1, wherein the ~~hash representation~~ numeric code is not a unique code for ~~said~~ the one syntactic

element, ~~said~~ the element being constrained, to a probability level, in terms of at least one of (i) a number of characters in the element and (ii) a permissible number of permutations of characters in the element.

8. (canceled).

9. (currently amended) A method according to claim ~~[[2]]~~ 1, wherein said processing step comprises a sub-step of:

determining an extended ~~hash representation~~ numeric code of both (i) ~~said~~ the one syntactic element being a first instance of ~~said~~ the first type, and (ii) another syntactic element being a second instance of ~~said~~ the first type, within which ~~said~~ first instance, ~~said~~ the second instance is nested.

10. (currently amended) A method according to claim 1 comprising, for another one of ~~said~~ the syntactic elements, the further steps of:

identifying a type of the other element; and if the type of the other element is equivalent to ~~said~~ the first type:

(i) processing the other element by applying the hash function thereto, thereby to generate thereby determine a second ~~hash representation~~ numeric code thereof from the other element; and

(ii) augmenting the enclosed ~~said at least partial structural~~ representation of the markup language document using the second numeric code ~~hash representation~~, wherein:

said processing of the one element and said ~~second~~ processing of the other element ensure that if a first relationship exists between the one element and the other element, then a second relationship which is representative of the first relationship, exists between the ~~hash representation~~ numeric code of the one element and the ~~hash representation~~ second numeric code of the other element.

11. (currently amended) A method according to claim 10, wherein:

- the one element is a start tag;
- the other element is an end tag;
- the ~~hash representation~~ numeric code of the one element is a corresponding hashed start tag, and[[]]
- the second ~~hash representation~~ numeric code of the other element is a corresponding hashed end tag.

12. A method according to claim 11, wherein:

- the end tag is a first modification of the start tag; and
- the hashed end tag is a second modification of the hashed start tag, ~~said the~~ second modification being representative of the first modification.

13. (original) A method according to claim 12, wherein:

- the end tag is the same as the start tag apart from having a distinguishing character incorporated therein; and
- the hashed end tag is at least one of:

the same as the hashed start tag;

the same as the hashed start tag apart from having a distinguishing character incorporated therein; and

the hashed start tag having been processed by an operator.

14. (currently amended) A method according to claim ~~[[12]]~~ 10, wherein:

the one element and the other element comprise respectively a start tag and an end tag, being a first pair of tags;

corresponding hashed start and end tags for ~~said the~~ first pair of tags are incorporated into the encoded ~~partial-structural~~ representation of ~~said the~~ document;

the document further includes a second pair of tags comprising a respective start tag and end tag, ~~said the~~ second pair of tags being nested within ~~said the~~ first pair of tags in ~~said the~~ document, and said method comprising further steps of:

processing ~~said the~~ second pair of tags to form corresponding second hashed start and end tags;

augmenting ~~said at least partial-structural~~ the encoded representation of the document using ~~said the~~ corresponding second hashed start and end tags so that ~~said the~~ second hashed start and end tags indicate a nesting in relation to ~~said the~~ hashed start and end tags for the first pair of tags which is equivalent to the nesting of ~~said the~~ second pair of tags within ~~said the~~ first pair of tags.

15. A method according to claim 14 comprising, prior to said augmenting step, a further step of:

concatenating the first hashed start tag with the second hashed start tag, and concatenating the first hashed end tag with the second hashed end tag, to thereby form respective extended hashed start and end tags for said second pair,

wherein[[:]] said augmenting step is performed using ~~said~~ the respective extended hashed start and end tags for ~~said~~ the second pair, and[[:]]

~~said~~ the extended hashed start and end tags indicate a nesting in relation to ~~said~~ the hashed start and end tags for the first pair of tags which is equivalent to the nesting of ~~said~~ the second pair of tags within ~~said~~ the first pair of tags.

16. (currently amended) A method according to claim 1, wherein ~~the~~ said augmenting step is succeeded by a well-formedness checking step against a syntactic rule, said well-formedness checking step comprising checking ~~said at least partial structural representation of~~ the markup language document against the syntactic rule by numerically comparing corresponding numeric code ~~hashed representations~~ of elements in ~~said at least partial structural~~ the encoded representation of the markup language document.

17. (currently amended) A method according to claim 16, wherein said numerically comparing step is succeeded by a further step of:

string-comparing, in accordance with ~~said~~ the syntactic rule, corresponding ~~non-processed hashed~~ representations of elements not of ~~said~~ the first type.

18. and 19. (canceled).

20. (currently amended) A method according to claim 14, comprising a further step of:

checking the well-formedness of ~~said at least partial structural~~ the encoded representation of the document against a syntactic rule.

21. (currently amended) A method according to claim 20, wherein the syntactic rule relates to proper nesting of tags and said checking step comprises sub-steps of:

performing a numerical comparison across hashed tags in ~~said at least partial structural~~ the encoded representation of the document, thereby to ~~thereby~~ identify ~~said~~ the first hashed start and end tags and ~~said~~ the second hashed start and end tags; and
verifying that the second hashed start and end tags indicate a proper nesting in relation to ~~said~~ the first hashed start and end tags.

22. (currently amended) A method according to claim 21, wherein the numerical comparison is followed by a further step of:

performing a string comparison, in accordance with ~~said~~ the syntactic rule, across non-processed ~~non-hashed~~ parts of respective tags in ~~said at least partial structural~~ the encoded representation of the document .

23. (currently amended) A method according to claim 15, comprising a further step of:

checking the well-formedness of ~~said at least partial structural~~ the encoded representation of the document against a syntactic rule.

24. (currently amended) A method according to claim 23, wherein the syntactic rule relates to proper nesting of tags and said checking step comprises sub-steps of:

performing a numerical comparison across hashed tags in ~~said at least partial structural~~ the encoded representation of the document, thereby to ~~thereby~~ identify ~~said~~ the first hashed start and end tags and ~~said~~ the extended hashed start and end tags; and
verifying that the extended hashed start and end tags indicate a proper nesting in relation to ~~said~~ the first hashed start and end tags.

25. (currently amended) A method according to claim 24, wherein the numerical comparison is followed by a further step of:

performing a string comparison across non-hashed parts of respective tags in ~~said at least partial structural~~ the encoded representation of the document.

26. (currently amended) A method according to claim 16, wherein ~~the~~ said well-formedness checking step is one of (a) succeeded by, (b) included in, and (c) replaced by a validation step against a validation reference document VRD, said validation step comprising sub-steps of:

(a) processing the VRD, said processing comprising, for a syntactic element in the VRD, sub-sub-steps of:

(i) identifying a type of ~~said~~ the syntactic element of the VRD;
and

(ii) processing the syntactic element by ~~applying~~ determining a hash ~~function thereto~~ representation thereof if ~~said~~ the type is ~~said~~ the first type, ~~the hash function generating a numeric code from the element~~; and

(b) checking ~~said at least partial structural~~ the encoded representation of the markup language document against the processed VRD, said checking comprising a sub-sub-step of numerically comparing corresponding numeric codes ~~hashed~~ representations of the elements.

27. and 28. (canceled).

29. (currently amended) A method according to claim 26, wherein said numerically comparing step is succeeded by a further step of string-comparing corresponding non-processed ~~non-hashed~~ representations of elements not of ~~said~~ the first type.

30. A method according to claim 26, wherein ~~said~~ the first type is one of:

one of a structural element and a part thereof;

a definition of ~~said~~ the structural element;

a declaration of ~~said~~ the structural element; and

a match of ~~said~~ the structural element.

31. A method according to claim 30, wherein ~~said~~ the structural element is a tag.

32. and 33. (canceled).

34. (currently amended) An apparatus for generating an encoded representation of parsing a markup language document comprising syntactic elements, said apparatus comprising:

parsing means for parsing the markup language document to identify at least one syntactic element of that document;

identifying means for identifying a type of the element;

processing means for processing the element by applying ~~determining~~ a hash function thereto ~~representation thereof~~ if ~~said~~ the type is a first type, said hash function generating a numeric code from the element; and

generating ~~augmenting~~ means for generating ~~augmenting an at least partial structural representation of the document using the hash~~ the encoded representation including the numeric code if ~~said~~ the type is ~~said~~ the first type.

35. - 38. (canceled).

39. (currently amended) A computer program which is configured to make a computer execute a procedure to generate an encoded representation of ~~parse~~ a markup language document comprising syntactic elements, said program comprising:

code for parsing the markup language document to identify at least one syntactic element of that document;

code for identifying a type of ~~[[an]]~~ the identified element;

code for processing the identified element by ~~applying~~ determining a hash function thereto ~~representation thereof~~ if said the type is a first type, the hash function generating a numeric code from the identified element; and

code for generating ~~augmenting an at least partial structural representation of the document using the encoded hash representation including the numeric code~~ if said the type is said the first type.

40. - 43. (canceled).

44. (currently amended) A computer program product including a computer readable medium having recorded thereon a computer program which is configured to make a computer execute a procedure to generate an encoded representation of ~~[[parse]]~~ a markup language document comprising syntactic elements, said program comprising:

code for parsing the markup language document to identify at least one syntactic element of that document;

code for identifying a type of the element;

code for processing the element by applying determining a hash function thereto representation thereof if said the type is a first type, the hash function generating a numeric code from the element; and

code for generating augmenting an at least partial structural representation of the document using the encoded hash representation including the numeric code if said the type is said the first type.

45. and 46. (canceled).

47. (currently amended) An encoded at least partial structural representation of a markup language document comprising syntactic elements, said at least partial the encoded representation having been produced by a method comprising, for one of said syntactic elements, the steps of:

parsing the markup language document to identify at least one the syntactic element of that document;

identifying a type of the element;

processing the element by applying determining a hash function thereto representation thereof if said the type is a first type, the hash function generating a numeric code from the element; and

generating augmenting an at least partial structural representation of the document using the encoded hash representation including said numeric code if said the type is said the first type.

48. (currently amended) An apparatus for generating an encoded representation of ~~parsing~~ a markup language document comprising syntactic elements, said apparatus comprising:

a processor;

a memory for storing (i) the document, and (ii) a program which is configured to make the processor execute a procedure to generate the encoded representation, ~~parse the document~~;

said program comprising:

(i) code for parsing the markup language document to identify at least one syntactic element of that document;

(ii) code for identifying a type of ~~[[an]]~~ the element;

(iii) code for processing the element by ~~applying~~ determining a hash function thereto ~~representation thereof~~ if ~~said~~ the type is a first type, the hash function generating a numeric code from the element; and

~~((iv))~~ ~~[[ivii]])~~ code for generating ~~augmenting an at least partial structural representation of the document using the~~ encoded hash representation including said numeric code if ~~said~~ the type is ~~said~~ the first type.

49. - 58. (canceled).

59. (new) A method according to claim 9, further comprising the step of validating the markup language document against a validation reference document (VRD), said validating comprising sub-steps of:

(a) processing the markup language document, for each document tag identified therein, if the document tag is not a first document tag in a corresponding markup language document tag hierarchy, said processing comprising the sub-sub-steps of:

- (i) determining a hierarchy position of the document tag;
- (ii) determining an extended numeric code of the document tag concatenated with a numeric code of a previous document tag in the document tag hierarchy; and
- (iii) storing the extended numeric code of the document tag if the document tag is more deeply nested than a previous document tag;

(b) processing the VRD, for each tag identified therein, if the tag is not a first tag in a corresponding tag hierarchy, said processing comprising sub-sub-steps of:

- (i) determining a hierarchy position of the tag;
 - (ii) determining an extended numeric code of the tag concatenated with a numeric code of a previous tag in the corresponding tag hierarchy; and
 - (iii) storing the extended numeric code of the tag in a list; and
- (c) validating the markup language document if the extended numeric code of the document tag is one of found in the list and is a valid subset of a member of the list.

60. (new) A method according to claim 1, further comprising the step of determining a compressed representation of the syntactic element if the type is not the first type.

61. (new) A method of decoding an encoded representation of a markup language document comprising encoded syntactic elements, the encoded representation having been encoded according to the method of claim 1, said decoding method comprising the steps of:

identifying an encoded element from the encoded representation;

processing the encoded element by one of:

(i) applying an inverse hash function, complementing the hash function of claim 1, to the encoded element, thereby to generate a syntactic element if the encoded element is a numeric code; and

(ii) decompressing the encoded element if the encoded element is not a numeric code; and

retaining the syntactic element in the document.

62. (new) An apparatus according to claim 34, wherein each of said means forms a part of an embedded computer system.